

# OPTIMIZING OPERATION OF A HIERARCHICAL CAMPUS-WIDE MOBILE GRID

Konstantinos Katsaros  
Mobile Multimedia Laboratory  
Department of Computer Science  
Athens University of Economics and Business  
Athens, Greece

George C. Polyzos  
Mobile Multimedia Laboratory  
Department of Computer Science  
Athens University of Economics and Business  
Athens, Greece

## ABSTRACT

Recent advances in mobile communications and computing and strong interest of the scientific community in the *Grid* have led to research into the *Mobile Grid*. We discuss various approaches proposed in the literature and try to point out the fundamental issues and problems emerging from the introduction of mobile devices and wireless communications in the context of the Grid computing paradigm. We further describe an architecture for the realization of a Mobile Grid and investigate key design decisions and optimizations.

## I. INTRODUCTION

Grid computing has emerged as a paradigm for the coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations [1]. A grid computing system is essentially a large-scaled distributed system designed to aggregate resources from multiple sites. On the other hand, the relatively recent advantages in mobile and wireless communications have resulted in the availability of an enormous number of mobile computing devices such as laptop PCs and PDAs.

In effect, it is considered natural to extend the idea of resource sharing to mobile and wireless communication environments. We believe that since the number of available mobile devices is nowadays enormous and their computational power is constantly increasing, the aggregated sum of their resources should be exploited. Especially in environments such as university campuses, the number of available mobile devices is in the order of thousands [2]. Considering the fact that such networking environments are usually under a single network administration, we believe that accumulating these concentrated and potentially unexploited resources is an interesting and significant challenge.

In this paper, we discuss a campus-wide hierarchical Mobile Grid system architecture in which mobile nodes (MNs), willing to offer their computational resources, move between WLANs [3]. This willingness is based on *reciprocity*. We consider *divisible load* [4] applications where the load of computation can be divided into several parts i.e. a job that can be divided into tasks that can be carried out independently of each other. In order to investigate the feasibility and gain insights on the performance of the proposed architecture we utilized the WLAN traces collected at Dartmouth College [2]. These traces provided a realistic simulation environment suitable for the study of some key performance aspects of a hierarchical campus-wide Mobile Grid system.

The remainder of the paper is organized as follows. In Section II. we examine related work. In section III. we describe the networking architecture and discuss the incentives mechanism, while in section IV. we present the framework for the evaluation of the proposed system. In section V. we present the results of the performance evaluation. We finally conclude in section VI. and present our plans for future work.

## II. RELATED WORK

There are various, quite different approaches on the exact character of a potential Mobile Grid system.

### A. Mobile Devices as Resource Consumers

In this case [5, 6, 7, 8], focus is given on the fact that mobile devices have limited computational and/or storage capabilities compared to their stationary counterparts. The grid can provide the resources missing in mobile devices on demand. Nevertheless, problems related to the very nature of wireless and mobile devices arise here. Submitting jobs and receiving the results back is not as straightforward as it may seem, since power constraints and frequent disconnections are prevalent in wireless and mobile communications. The use of proxies is proposed in [5] which act as gateways to the grid. These proxies undertake the role of the mediator between the mobile device and the grid system, and try to hide the instability of the wireless/mobile environment by acting on behalf of the mobile device.

Other approaches on solving the resource limitation problem of mobile devices target at providing a "smart" environment for *pervasive* computing [6, 7]. In these approaches, mobile devices are considered as pure access devices without need for enhanced processing and/or storage capabilities. The role of the grid is to provide all the functionalities required by users pushing this way the complexity of the whole system to the networking environment rather than to the edges.

### B. Mobile Devices as Resource Providers

In this case [9, 10, 5, 11, 12], strong emphasis is given to two important factors. First, even though mobile devices have limited resources compared to their stationary counterparts, they seem to increasingly gain sufficiently powerful CPUs and storage means. In effect, they are considered capable of providing useful resources. Second, since the number of mobile devices continuously increases, the aggregate of their resources cannot be considered negligible [10]. However, again, mobility related problems such as limited power and intermittent con-

nectivity pose significant difficulties in the effort of exploiting these resources.

Two fundamentally different architectures have been followed in an effort to exploit resources relying in mobile devices: mobile grids on-site and mobile ad-hoc grids.

### 1) Mobile Grids On-Site

In the case of creating mobile grids on-site [10, 5, 11, 13], mobile devices residing in a well defined area, such as a cell in cellular networks or a WLAN hot-spot, (Service Area, SA) are co-ordinated by a central entity (residing at the Access Point/Base Station, BS) in order to perform a task (computation grid). The devices provide the description of their capabilities and the degree of their availability to the BS. The BS is then responsible for decomposing an incoming request and scheduling the overall execution by providing specific tasks to each of the participating mobile devices. A request can either come from a mobile device in the SA or another client outside this area (fixed or mobile). The advantage of this approach is that the BS can act as a mediator capable of hiding the heterogeneity of the participating devices from the requesting node, coordinating the overall execution of the submitted job and even allowing the grid system to appear to the rest of the network as an ordinary grid node [10].

### 2) Mobile ad-hoc grids

In the case of mobile ad-hoc grids [12, 14], there is no central authority responsible for the co-ordination of the overall job execution. Further problems arise in this case due to the ad-hoc nature of such systems. The absence of central coordination imposes difficulties in service discovery, job scheduling and monitoring. An approach towards overcoming this limitation, is the formation of a virtual backbone consisting of a number of, possibly more powerful, mobile nodes responsible of coordinating the mobile nodes residing in a certain area of the overall ad-hoc network[12, 14]. The instability of the network topology induces further difficulties due to unique ad-hoc related characteristics such as network partitioning and multi-hop routing.

### C. The Akogrimo Integrated project case

The EU IST project Akogrimo "Access to Knowledge through the Grid in a Mobile World" [15] aims at the provision of a framework for deploying mobile grids focused on solving complex problems by exploiting knowledge as their main input. The distinction between this type of mobile grid and the ones described above is that the resource shared in this grid is *information*. In Akogrimo, data and computational power are accessed by dedicated fixed-grid providers [16].

## III. MOBILE GRID ARCHITECTURE

In this paper, we refer to a hierarchical campus-wide Mobile Grid architecture [3] following the Mobile Grid On-Site approach as described in 1). The resource shared is computational power. In a typical university campus, students, faculty and administrative personnel move with their portable devices

between multiple WLANs deployed in the campus buildings. As shown in Fig. 1, at each level of the hierarchy a Mobile Grid Scheduler (MGS) is responsible for receiving jobs from the upper layer MGS, splitting them into separate tasks and assigning these tasks to the lower layer MGSs. Each MGS is also responsible for communicating the available resources up to the hierarchy of MGSs. An intermediate MGS (I-MGS) located at each building can receive a job from the root MGS (R-MGS) and assign the produced tasks to the local schedulers located at each WLAN in the building (L-MGS). Each L-MGS further splits the received task and assigns the resulting sub-tasks to the MNs residing at the WLAN it serves. Participating MNs process the received input data and return the results back to the assigning L-MGS. A MN may submit a job directly to the L-MGS at its WLAN.

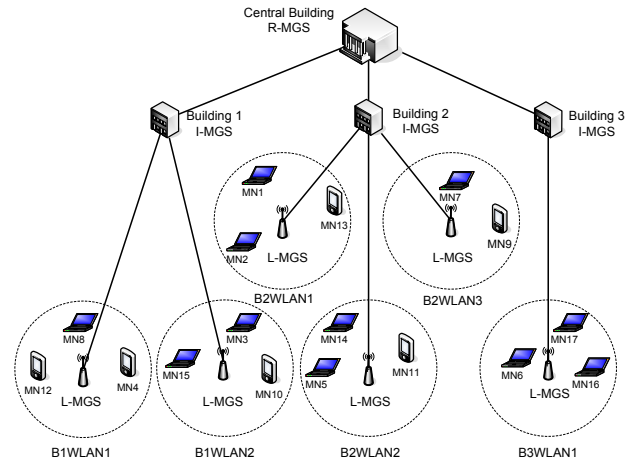


Figure 1: Hierarchical Campus-Wide Mobile Grid architecture.

### A. Incentives

A critical issue in the deployment of the proposed architecture concerns the incentives given to mobile nodes in order to motivate them to offer their resources.

As mentioned earlier, our scheme is based on *reciprocity*. A MN can submit a job to the Mobile Grid and take advantage of its computational resources only if it shares its own resources as well. Obviously, the amount of resources offered by a MN to the Mobile Grid over a certain period of time may not exceed the amount of resources required for the execution of its own jobs during the same time-period. Otherwise a MN will simply prefer not to participate in the Mobile Grid and compute its own jobs directly. Therefore, the main target is to exploit the vast number of MNs by harvesting relatively small amounts of resources from each of them at a time and taking advantage of the parallel character of task execution.

To this end, we propose a simple mechanism in which each MN collects *credits* for each task it processes. The value of each credit is proportional to the size of the corresponding task. A job may be submitted by a MN only when its *credit*-value is less or equal the total value of all collected credits. A MN may choose the time and the extend (i.e. preferable task size) to

which it will offer its resources (and collect the corresponding credits) to the Mobile Grid based on several criteria such as available power and current device workload. Therefore, the credits required for the submission of a job may be collected at will and over a period of time which will ensure that the MN will not be disturbed by the energy and computational power consumption.

Further investigation towards the precise design of the incentives scheme together with an intelligent *accounting mechanism* is subject to imminent work.

#### IV. EVALUATION FRAMEWORK

##### A. Formulation of the problem

The core functionality of an L-MGS is to receive a job (or a task from an ancestor MGS) and, in the context of *divisible load* applications [4], divide the submitted workload into tasks (or sub-tasks) for submission to the MNs residing at the corresponding WLAN. The whole process consists of three distinct steps: the transfer of the input workload to the MN, the task execution and the return of the results back to the L-MGS. In the absence of disconnection events, each step requires  $T_{IN}$ ,  $T_{EXEC}$ ,  $T_{OUT}$  amount of time to complete, respectively. We also define  $T_{TOTAL}$  as:

$$T_{TOTAL} = T_{IN} + T_{EXEC} + T_{OUT} \quad (1)$$

The *Communication to Computation Ratio* (CCR) [17] of a divisible load application is defined as :

$$CCR = \frac{CommunicationCost}{ComputationCost} \quad (2)$$

The *CommunicationCost* factor denotes the time required by the MN to receive the task workload. The *ComputationCost* denotes the time required by the MN to complete the execution of the submitted task.

##### B. Performance Constraints

The performance of the proposed architecture heavily depends on the *Response Time* ( $RT$ ) of each MN working on a task i.e. the time required for the MN to return the results back to the L-MGS. In ideal environments, where no disconnections take place, we obviously have  $RT = T_{TOTAL}$ . However, in a wireless and mobile context, intermittent connectivity introduces further delays on the completion of the whole process. The process of transferring the input/output data to/from the MN may be interrupted by a disconnection event due to the MN's mobility and/or power constraints. Hence, a further delay on  $T_{IN}$  and/or  $T_{OUT}$  is imposed in this case. It is important to note that disconnections do not affect the time required for the MN to process the input data since a MN may continue processing even if it is disconnected. This time is affected however by user actions, since mobile devices are strictly personal (e.g. a user may decide to turn off his device). In this case, an additional delay is imposed on  $T_{EXEC}$ . This delay is not investigated in the remainder of this paper.

The authors in [18] have concluded that, for the same collection of traces utilized in this paper, the average length of a

MN *session*<sup>1</sup> is 16.6 minutes, 71 % of sessions finish in less than an hour and 27% of the sessions last less than one minute. These findings clearly demonstrate the intermittent character of these MNs' connectivity in the networking environment of our architecture.

##### C. Disconnection and Rescheduling

As discussed above, intermittent connectivity may impose a delay overhead on the completion of a task execution. Therefore, the resulting question is how to overcome this barrier in order to achieve high performance in terms of RT. In this context, a fundamental design decision is whether a task executed by a mobile node must be aborted upon disconnection, in the sense that it will be rescheduled (termed as *Abort-Reschedule* policy), or it must be carried out by the mobile node until a new point of attachment to the network is found and the results can be returned back (termed as *Carry* policy).

In the architectures presented in [11] and [13], the first approach is followed in view of the latency incurred by a hand-over. We believe that this issue requires further investigation. The computational and energy resources spent by a MN for the execution of a task are obviously wasted if the assigned task is aborted in the event of a disconnection. From a system designer point of view, this inevitably leads to the underutilization of the available resources. From a MN point of view, this fact leads to user annoyance unless a proper accounting mechanism ensures that he/she will indeed receive the credits for the resources offered until task abortion.

On the other hand, the *Abort-Reschedule* policy may lead to significant performance gains in terms of response time (See Section B.). Specifically, the final response time for an assigned task is defined as  $RT_{Abort}$  in the case of the *Abort-Reschedule* policy and as  $RT_{Carry}$  in the case of the *Carry* policy. Once a disconnection event has occurred, we have:

$$RT_{Abort} = T_{Complete} + T_{Arrival} + RT'_{Abort} \quad (3)$$

$$RT_{Carry} = T_{TOTAL} + T_{Disconnection} \quad (4)$$

, where  $T_{Complete}$  is the portion of  $T_{TOTAL}$  that has actually been completed until the MN gets disconnected,  $T_{Arrival}$  is the time the L-MGS waits for a MN to enter its area of responsibility i.e. WLAN, in order to re-assign the aborted task to it,  $RT'_{Abort}$  is the response time of this new MN and  $T_{Disconnection}$  is the time spent in disconnection by the MN. Obviously,  $T_{Arrival} = 0$  when a new MN i.e. a MN that has not received any task, has entered the WLAN before the disconnection event has been realized. Furthermore,  $T_{Disconnection}$  corresponds to the aggregate time spent in disconnection. For simplicity, we assume that an L-MGS is immediately aware of the disconnection event.

Obviously, if  $RT_{Abort} < RT_{Carry}$ , rescheduling a task results in lower  $RT$  and consequently in an enhanced system performance. Otherwise, the performance of the system degrades

<sup>1</sup>A *session* is defined as the period of time in which a user (car d) joins the network, uses the network, possibly roams to other APs in the same subnet, and leaves the network.

due to the *Abort-Reschedule* policy. Therefore, the choice between the two policies heavily depends on the aforementioned MNs connectivity characteristics i.e. the inter-arrival time, the disconnection time and the probability of disconnection which affects the depth of the recursive equation (3) .

It is stressed at this point that the aforementioned policies are not considered as the primary means for addressing the problem of intermittent connectivity. Other techniques such as *load balancing*, exploitation of *connectivity profiles* and task replication [3] can be employed in order to smooth the effects of they instable networking environment. Moreover, it is noted that these policies are not mutually exclusive. For example, building connectivity profiles for each MN may assist an intelligent scheduler in choosing between these approaches with respect to each MN's connectivity characteristics. For instance, if a MN tends to get disconnected for very small periods of time, it is desirable to follow the *Carry* policy since the MN is expected to get connected soon and return the results back in less time than in the *Abort-Reschedule* approach.

In the next section we present our first findings for these two policies based on a series of emulations.

## V. PERFORMANCE EVALUATION

In order to study the performance of the proposed architecture we need to define suitable metrics. We chose to study the resulting  $RT$ s for various  $T_{IN}$ ,  $CCR$  and  $T_{OUT}$  values. Instead of providing explicit values for the input and output loads (e.g. in MB) we chose to describe these factors in terms of *time* i.e. the time required for the transfer of input and output data in the absence of disconnection events. Our decision is based on the fact that we do not have any information on the actual throughput in the WLANs under study. An indication of the actual input and output loads may be derived from performance measurement studies. In [19], the measurements show a net throughput of 47% at 11Mbit/s, including the TCP/IP protocol stack overhead.

### A. Traces

In order to investigate the performance of the proposed architecture under realistic conditions we utilized the WLAN traces available at [2]. The traces were collected from April 2001 to March 2003. Due to holes in the traces during fall of 2001 we only used data collected during 2002 and 2003. These traces contain information for 5982 distinct wireless cards and 566 APs in 166 buildings.

We extracted approximately 1000 *testing environments*. Each testing environment corresponds to a certain time and AP in the traces and refers to the MNs residing in the specific AP at the chosen point in time. All testing environments were uniformly extracted from the whole period of traces and the entirety of the APs.

### B. Abort-Reschedule Vs. Carry Policies

In this section we present preliminary results obtained from the utilized traces for the comparison of the two approaches. Fig. 2 presents the  $RT$  measured for various values of the input load

(i.e.  $T_{IN}$ ) and  $CCR = 1/10$ . We considered the case where  $T_{IN} = T_{OUT}$ . The resulting  $T_{TOTAL}$  is equally partitioned to all MNs residing in the WLAN at the time of scheduling. For comparison reasons, the figure also depicts the time required by a single MN to receive the input data, complete the computation of the task and return the results back to the L-MGS in the absence of disconnection events (Single Stationary curve). Similar results were obtained for  $CCR = 1$  and  $CCR = 1/5$  but are not shown here due to length limitations.

These results clearly demonstrate the superiority of the *Abort-Reschedule* policy in the networking environment under evaluation, regardless the  $CCR$  value. This implies that the impact of disconnection periods on the resulting  $RT_{Carry}$  is severe and surely outweighs the impact of weighting for a new MN arrival and rescheduling the task in the *Abort-Reschedule* policy. A more refined simulation-based investigation of these mobility/connectivity characteristics is subject to imminent work.

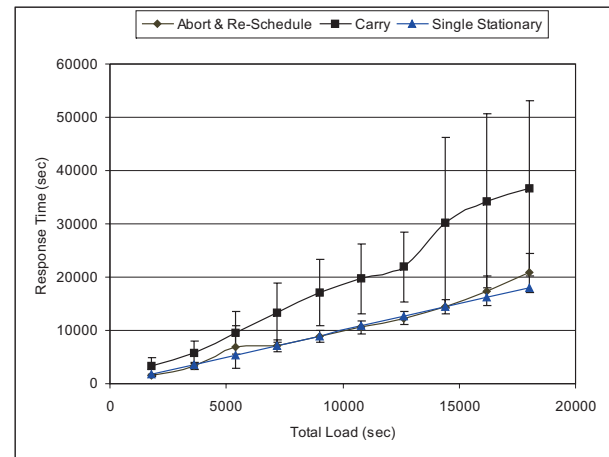


Figure 2: Response Time for Abort-Reschedule and Carry policies,  $CCR=1/10$

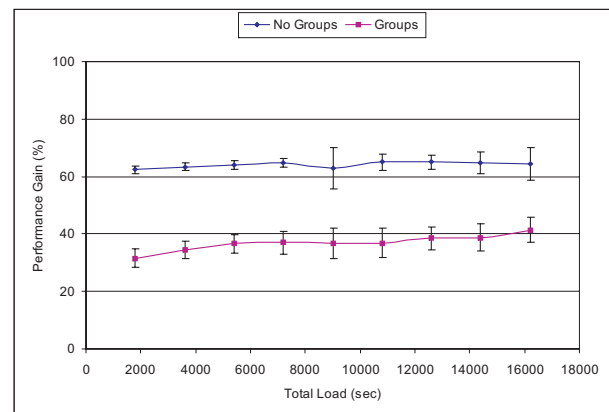


Figure 3: Task Replication Performance Gain

### C. Task Replication

In previous work [3] we investigated the approach of *task replication* i.e. the assignment of a certain task to more than one of the MNs residing in the same WLAN, in the context of the *Carry* policy (termed as *Groups* approach). This approach is based on the fact that not all MNs present the same networking behavior. Therefore, if the same task is submitted to multiple MNs it is highly probable that one of them will eventually return the results earlier than the others. The produced results clearly demonstrated the superiority of the *task replication* approach, in terms of response time, in the presence of disconnection events.

In Fig. 3, we present our findings for this task replication scheme in the context of the *Abort-Reschedule* policy. On average, the *Groups* approach leads to a performance gain of 37%, compared to the Single Stationary approach, while the *No Groups* approach i.e. not employing task replication, yields a performance gain of 64%. As expected, the task replication approach is not preferred in the context of the *Abort-Reschedule* policy because it results in an increased task size which in turn increases the probability of disconnection events and most importantly the final  $RT_{Abort}$  since  $RT_{Abort} \geq T_{TOTAL}$ .

In Fig. 4 we compare the resulting RTs for the *Carry* policy with task replication and the *Abort-Reschedule* with no task replication in order to examine which combination of the afore-discussed presents better performance. It is clear that the *No Groups/Abort-Reschedule* approach outperforms the *Groups/Carry* approach. Again, the increased task size in the *Groups/Carry* approach leads to a relatively increased response time.

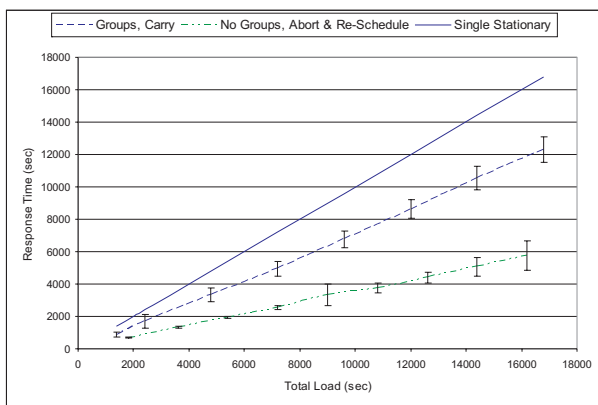


Figure 4: Comparison between *No Groups/Abort-Reschedule* and *Groups/Carry* policies

## VI. FUTURE WORK

In this paper, we investigated the fundamental issues rising in the path towards the realization of the Mobile Grid paradigm. We discussed various approaches in literature and pointed out the problems introduced by node mobility. Moreover, we studied the performance of a hierarchical, campus-wide networking architecture based on appropriate traces.

Our plans for imminent future work include, but are not limited to, the investigation and detailed design of the incentives mechanism. Furthermore we plan the study of load balancing techniques throughout the hierarchical structure of the Mobile Grid architecture based on the construction of suitable mobility patterns that will model the MNs' networking behavior.

## REFERENCES

- [1] I.T. Foster, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," Int. J. of Supercomputer Applications, 15(3), 2001.
- [2] D. Kotz, T. Henderson and I. Abyzov, "CRAWDAD trace set dartmouth/campus/movement (v. 2005-03-08)," Downloaded from <http://crawdad.cs.dartmouth.edu/dartmouth/campus/movement>, 2005.
- [3] K. Katsaros and G.C. Polyzos, "Optimizing Operation of a Hierarchical Campus-wide Mobile Grid for Intermittent Wireless Connectivity," IEEE LAN/MAN Workshop (LANMAN 2007), 2007.
- [4] V. Bharadwaj, D. Ghose, V. Mani and T. Robertazzi, "Scheduling Divisible Loads in Parallel and Distributed Systems," IEEE Computer Society Press, 1996.
- [5] S.-M. Park, Y.-B. Ko, and J.-H. Kim, "Disconnected Operation Service in Mobile Grid Computing," Int. Conf. on Service Oriented Computing (ICSOC'2003), Trento, Italy, Dec 2003
- [6] G. Banavar, J. Beck, E. Gluzberg, J. Munson, J. Sussman, and D. Zukowski, "Challenges: An Application Model for Pervasive Computing," 6th ACM MobiCom, Boston, MA, Aug 2000
- [7] S.H. Srinivasan, "Pervasive wireless grid architecture," 2nd Annual Conf. on Wireless On-demand Network Systems and Services (WONS), St. Moritz, Switzerland, 2005.
- [8] M. Migliardi, M. Maheswaran, B. Maniymaran, P. Card and F. Azzedin, "Mobile interfaces to computational, data, and service grid systems", ACM Mobile Computing and Communications Review, October 2002
- [9] A. Litke and D. Skoutas and T. Varvarigou, "Mobile grid computing: Changes and challenges of resource management in a mobile grid environment," Practical Aspects of Knowledge Management, Austria, December 2004.
- [10] T. Phan, L. Huang, and C. Dulan, "Challenge: Integrating Mobile Wireless Devices Into the Computational Grid," 8th ACM Int. Conf. on Mobile Computing and Networking (MobiCom '02), September 25-27, 2002, Atlanta, GA.
- [11] S. Kurkovsky and Bhagyavati, "Wireless Grid Enables Ubiquitous Computing," 16th Int. Conf. on Parallel and Distributed Computing Systems (PDCS-2003), Reno, NV, Aug. 2003.
- [12] H. Li, L. Sun and E.C. Ifeachor, "Challenges of Mobile ad-hoc Grids and their Applications in e-Healthcare," 2nd Int. Conf. on Computational Intelligence in Medicine and Healthcare (CIMED2005), Lisbon, Portugal
- [13] S. Kurkovsky, Bhagyavati, and A. Ray, "A collaborative problem-solving framework for mobile devices," Southeast Regional Conference (ACM-SE 2004), 2004.
- [14] D.C. Marinescu, G.M. Marinescu, Y.Ji and B. Ladislau, "Ad hoc grids: communication and computing in a power constrained environment," International Performance, Computing, and Communications Conference, pp. 113- 122, 9-11 April 2003
- [15] AKOGRIMO Integrated Project <http://www.akogrimo.org>
- [16] M. Waldburger and B. Stiller, "Toward the Mobile Grid: Service Provisioning in a Mobile Dynamic Virtual Organization," 4th ACS/IEEE Int. Conf. on Computer Systems and Applications (AICCSA-06), Dubai/Sharjah, UAE, March 2006, pages 579-583.
- [17] Y. Cardinale and H. Casanova, "An evaluation of Job Scheduling Strategies for Divisible Loads on Grid Platforms," High Performance Computing and Simulation Conference (HPCS 2006), 2006.
- [18] D. Kotz and K. Essien, "Analysis of a Campus-wide Wireless Network," Int. Conf. on Mobile Computing and Networking (MobiCom 2002), 2002.
- [19] A. Kamerman and G. Aben, "Throughput performance of wireless LANs operating at 2.4 and 5 GHz," Personal, Indoor and Mobile Radio Communications (PIMRC 2000), 2000.